Never Ending Language Learning

Tom Mitchell

Machine Learning Department
Carnegie Mellon University
We will never really understand learning until we build machines that
- learn many different things,
- from years of diverse experience,
- in a staged, curricular fashion,
- and become better learners over time.
NELL: Never-Ending Language Learner

Inputs:
• initial ontology (categories and relations)
• dozen examples of each ontology predicate
• the web
• occasional interaction with human trainers

The task:
• run 24x7, forever
• each day:
  1. extract more facts from the web to populate the ontology
  2. learn to read (perform #1) better than yesterday
NELL today

Running 24x7, since January, 12, 2010

Result:
  • KB with > 70 million candidate beliefs
  • learning to read
  • learning to reason
  • extending ontology
NELL Today

- eg. “diabetes”, “Avandia”, “tea”, “IBM”, “love” “baseball” “BacteriaCausesCondition” “kitchenItem” “ClothingGoesWithClothing” ...

Recently-Learned Facts

<table>
<thead>
<tr>
<th>instance</th>
<th>iteration</th>
<th>date learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>lollypops magazine is a kind of media</td>
<td>844</td>
<td>06–jun–2014</td>
</tr>
<tr>
<td>examples of critical thinking at work is a cognitive action</td>
<td>844</td>
<td>06–jun–2014</td>
</tr>
<tr>
<td>diesel is a chemical</td>
<td>844</td>
<td>06–jun–2014</td>
</tr>
<tr>
<td>epigonism is a socio–political term</td>
<td>844</td>
<td>06–jun–2014</td>
</tr>
<tr>
<td>nuclear engineering and radiological sciences is an academic field</td>
<td>844</td>
<td>06–jun–2014</td>
</tr>
<tr>
<td>adelaide is the capital city of the state or province south australia</td>
<td>849</td>
<td>24–jun–2014</td>
</tr>
<tr>
<td>water has color turquoise</td>
<td>845</td>
<td>11–jun–2014</td>
</tr>
<tr>
<td>electronic arts001 has acquired origin</td>
<td>848</td>
<td>21–jun–2014</td>
</tr>
<tr>
<td>citizens is a bank in virgin islands</td>
<td>847</td>
<td>17–jun–2014</td>
</tr>
<tr>
<td>patty murray represents the region washington</td>
<td>845</td>
<td>11–jun–2014</td>
</tr>
</tbody>
</table>
How does NELL work?
Semi-Supervised Bootstrap Learning

Learn which noun phrases are cities:

Paris
Pittsburgh
Seattle
Montpelier

San Francisco
Berlin
denial

anxiety
selfishness
London

mayor of arg1
live in arg1
arg1 is home of traits such as arg1

it's underconstrained!!
Key Idea 1: Coupled semi-supervised training of many functions

- hard (underconstrained)
- semi-supervised learning problem

much easier (more constrained)
semi-supervised learning problem
Type 1 Coupling: Co-Training, Multi-View Learning

Supervised training of 1 function:

\[
\text{Minimize: } \sum_{<np,person> \in \text{labeled data}} |f_1(np) - person|
\]
Type 1 Coupling: Co-Training, Multi-View Learning

Coupled training of 2 functions:

Minimize:
\[ \sum_{<np,person> \in \text{labeled data}} |f_1(np) - person| + \sum_{<np,person> \in \text{labeled data}} |f_2(np) - person| + \sum_{np \in \text{unlabeled data}} |f_1(np) - f_2(np)| \]
Type 1 Coupling: Co-Training, Multi-View Learning

[Blum & Mitchell; 98]
[Dasgupta et al; 01]
[Ganchev et al., 08]
[Sridharan & Kakade, 08]
[Wang & Zhou, ICML10]
NELL: Learned reading strategies

Mountain:
"volcanic crater of _" "volcanic eruption like _" "volcanic region of _" "volcano, called _" "volcano known as _" "volcano Mt _" "volcanoes including _" "volcanoes, like _" "volcanoes, such as _" "volcanoes such as _" "we've climbed _" "weather atop _" "weather station atop _" "West face of _" "West ridge of _" "white ledge in _" "white summit of _" "whole earth, is _" "winter ascents in _" "winter ascents to _" "world famous view of _" "you've just climbed _" "'crater" "'eruption" "'fothills" "Camp" "'s drug guide" "'s east ridge" "'s Face" "'s North Peak" "'s North Ridge" "'s southeast ridge" "'s summit caldera" "'s west ridge" "(D,DDD ft" "'climbing permits" "'climbing safari" "consult el diablo" "'cooking planks" "dominates the skyline" "dominating the scenery"

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Feature</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>mountain</td>
<td>LAST=peak</td>
<td>1.791</td>
</tr>
<tr>
<td>mountain</td>
<td>LAST=mountain</td>
<td>1.093</td>
</tr>
<tr>
<td>mountain</td>
<td>FIRST=mountain</td>
<td>-0.875</td>
</tr>
<tr>
<td>musicArtist</td>
<td>LAST=band</td>
<td>1.853</td>
</tr>
<tr>
<td>musicArtist</td>
<td>POS=DT_NNS</td>
<td>1.412</td>
</tr>
<tr>
<td>musicArtist</td>
<td>POS=DT_JJ_NN</td>
<td>-0.807</td>
</tr>
<tr>
<td>newspaper</td>
<td>LAST=sun</td>
<td>1.330</td>
</tr>
<tr>
<td>newspaper</td>
<td>LAST=university</td>
<td>-0.318</td>
</tr>
<tr>
<td>newspaper</td>
<td>POS=NN_NNS</td>
<td>-0.798</td>
</tr>
<tr>
<td>university</td>
<td>LAST=college</td>
<td>2.076</td>
</tr>
<tr>
<td>university</td>
<td>PREFIX=uc</td>
<td>1.999</td>
</tr>
<tr>
<td>university</td>
<td>LAST=state</td>
<td>1.992</td>
</tr>
<tr>
<td>university</td>
<td>LAST=university</td>
<td>1.745</td>
</tr>
<tr>
<td>university</td>
<td>FIRST=college</td>
<td>-1.381</td>
</tr>
<tr>
<td>visualArtMovement</td>
<td>SUFFIX=ism</td>
<td>1.282</td>
</tr>
<tr>
<td>visualArtMovement</td>
<td>PREFIX=journal</td>
<td>0.234</td>
</tr>
</tbody>
</table>

Predicate | Web URL | Extraction Template |
|-----------|---------|---------------------|
Type 1 Coupling: Co-Training, Multi-View Learning

[Blum & Mitchell; 98]
[Dasgupta et al; 01 ]
[Ganchev et al., 08]
[Sridharan & Kakade, 08]
[Wang & Zhou, ICML10]
Multi-view, Multi-Task Learning

[Blum & Mitchell; 98]  
[Dasgupta et al; 01 ]  
[Ganchev et al., 08]  
[Sridharan & Kakade, 08]  
[Wang & Zhou, ICML10]  
[Taskar et al., 2009]  
[Carlson et al., 2009]  

NP: athlete(NP) \rightarrow person(NP)  
NP: athlete(NP) \rightarrow \text{NOT} sport(NP)  
NP: \text{NOT} athlete(NP) \leftarrow sport(NP)
Type 3 Coupling: Learning Relations

[Diagram showing relationships between NP1 and NP2 with nodes labeled as follows:
- playsSport(a,s)
- playsForTeam(a,t)
- teamPlaysSport(t,s)
- coachesTeam(c,t)]
Type 3 Coupling: Argument Types

playsSport(NP1,NP2) → athlete(NP1), sport(NP2)

over 2500 coupled functions in NELL
Initial NELL Architecture

Knowledge Base (latent variables)

- Beliefs
- Candidate Beliefs

Evidence Integrator

Continually Learning Extractors

- Text Context patterns (CPL)
- HTML-URL context patterns (SEAL)
- Morphology classifier (CML)
- Human advice
If coupled learning is the key, how can we get new coupling constraints?
Key Idea 2:

Discover New Coupling Constraints

- first order, probabilistic horn clause constraints:
  
  $0.93 \quad \text{athletePlaysSport}(?x,?y) \leftarrow \text{athletePlaysForTeam}(?x,?z) \quad \text{teamPlaysSport}(?z,?y)$
  
  - learned by data mining the knowledge base
  - connect previously uncoupled relation predicates
  - infer new unread beliefs
  - modified version of FOIL [Quinlan]
Learned Probabilistic Horn Clause Rules

0.93 \( \text{playsSport}(?x,?y) \leftarrow \text{playsForTeam}(?x,?z), \text{teamPlaysSport}(?z,?y) \)
Inference by KB Random Walks

[If: $x_1$ competes with $(x_1, x_2)$ then: economic sector $(x_2, x_3)$]

Then: economic sector $(x_1, x_3)$
CityLocatedInCountry(Pittsburgh) = ?

[Lao, Mitchell, Cohen, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

Pittsburgh

Pennsylvania

Feature = Typed Path
CityInState, CityInState^{-1}, CityLocatedInCountry

Feature Value

Logistic Regression
Weight
0.32

[Lao, Mitchell, Cohen, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

[Feature = Typed Path]

CityInState, CityInState⁻¹, CityLocatedInCountry

[Feature Value]

Logistic Regression Weight

0.32

[Lao, Mitchell, Cohen, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

Feature = Typed Path
CityInState, CityInState\(^{-1}\), CityLocatedInCountry

Feature Value

Logistic Regression
Weight
0.32
CityLocatedInCountry(Pittsburgh) = ?

\[ \Pr(U.S. \mid \text{Pittsburgh, TypedPath}) \]

**Feature = Typed Path**

CityInState, CityInstate\(^{-1}\), CityLocatedInCountry

**Feature Value**

0.8

**Logistic Regression Weight**

0.32

[Lao, Mitchell, Cohen, *EMNLP 2011*]
CityLocatedInCountry(Pittsburgh) = ?

[Lao, Mitchell, Cohen, EMNLP 2011]

<table>
<thead>
<tr>
<th>Feature = Typed Path</th>
<th>Feature Value</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityInState, CityInstate⁻¹, CityLocatedInCountry</td>
<td>0.8</td>
<td>0.32</td>
</tr>
<tr>
<td>AtLocation⁻¹, AtLocation, CityLocatedInCountry</td>
<td></td>
<td>0.20</td>
</tr>
</tbody>
</table>
Feature = Typed Path

CityInState, CityInState\(^{-1}\), CityLocatedInCountry
AtLocation\(^{-1}\), AtLocation, CityLocatedInCountry

Feature Value
0.8

Weight
0.32
0.20

CityLocatedInCountry(Pittsburgh) = ?

[Laos, Mitchell, Cohen, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

Feature = Typed Path
CityInState, CityInState^{-1}, CityLocatedInCountry
AtLocation^{-1}, AtLocation, CityLocatedInCountry

Feature Value
0.8

Weight
0.32
0.20

[Laó, Mitchell, Cohen, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

[Lao, Mitchell, Cohen, EMNLP 2011]

**Feature = Typed Path**

- CityInState, CityInstate\(^{-1}\), CityLocatedInCountry
- AtLocation\(^{-1}\), AtLocation, CityLocatedInCountry

**Feature Value**  | **Weight**
--- | ---
0.8 | 0.32
0.6 | 0.20
CityLocatedInCountry(Pittsburgh) = ?

Feature = Typed Path
CityInState, CityInState\(^{-1}\), CityLocatedInCountry
AtLocation\(^{-1}\), AtLocation, CityLocatedInCountry
...

Feature Value
CityLocatedInCountry(Pittsburgh) = U.S.  p=0.58

Logistic Regression
Weight

[Source: Lao, Mitchell, Cohen, EMNLP 2011]
CityLocatedInCountry(Pittsburgh) = ?

Feature Value
CityLocatedInCountry(Pittsburgh) = U.S. p=0.58

Feature = Typed Path
CityInState, CityInState^-1, CityLocatedInCountry
AtLocation^-1, AtLocation, CityLocatedInCountry
...

Feature Value
CityLocatedInCountry
0.8                          0.32
...
...
...

Weight
0.6                          0.20
...
...

1. Tractable (bounded length)
2. Anytime
3. Accuracy increases as KB grows
4. combines probabilities from different horn clauses

[Lao, Mitchell, Cohen, EMNLP 2011]
Random walk inference: learned rules

CityLocatedInCountry\((city, country)\):

8.04 cityliesonriver, cityliesonriver\(^{-1}\), citylocatedincountry
5.42 hasofficeincity\(^{-1}\), hasofficeincity, citylocatedincountry
4.98 cityalsoknownnas, cityalsoknownnas, citylocatedincountry
2.85 citycapitalofcountry, citylocatedincountry\(^{-1}\), citylocatedincountry
2.29 agentactsinlocation\(^{-1}\), agentactsinlocation, citylocatedincountry
1.22 statehascapital\(^{-1}\), statelocatedincountry
0.66 citycapitalofcountry

7 of the 2985 learned rules for CityLocatedInCountry
Key Idea 3:
Automatically extend ontology
Ontology Extension (1) [Mohamed et al., EMNLP 2011]

Goal:
• Add new relations to ontology

Approach:
• For each pair of categories C1, C2,
  • cluster pairs of known instances, in terms of text contexts that connect them
## Example Discovered Relations

[Mohamed et al. *EMNLP 2011*](#)

<table>
<thead>
<tr>
<th>Category Pair</th>
<th>Frequent Instance Pairs</th>
<th>Text Contexts</th>
<th>Suggested Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MusicInstrument Musician</td>
<td>sitar, George Harrison tenor sax, Stan Getz trombone, Tommy Dorsey vibes, Lionel Hampton</td>
<td>ARG1 master ARG2 ARG1 virtuoso ARG2 ARG1 legend ARG2 ARG2 plays ARG1</td>
<td>Master</td>
</tr>
<tr>
<td>Disease Disease</td>
<td>pinched nerve, herniated disk tennis elbow, tendonitis blepharospasm, dystonia</td>
<td>ARG1 is due to ARG2 ARG1 is caused by ARG2</td>
<td>IsDueTo</td>
</tr>
<tr>
<td>CellType Chemical</td>
<td>epithelial cells, surfactant neurons, serotonin mast cells, histamine</td>
<td>ARG1 that release ARG2 ARG2 releasing ARG1</td>
<td>ThatRelease</td>
</tr>
<tr>
<td>Mammals Plant Plant</td>
<td>koala bears, eucalyptus sheep, grasses goats, saplings</td>
<td>ARG1 eat ARG2 ARG2 eating ARG1</td>
<td>Eat</td>
</tr>
<tr>
<td>River City</td>
<td>Seine, Paris Nile, Cairo Tiber river, Rome</td>
<td>ARG1 in heart of ARG2 ARG1 which flows through ARG2</td>
<td>InHeartOf</td>
</tr>
</tbody>
</table>
NELL: sample of self-added relations

- athleteWonAward
- animalEatsFood
- languageTaughtInCity
- clothingMadeFromPlant
- beverageServedWithFood
- fishServedWithFood
- athleteBeatAthlete
- athleteInjuredBodyPart
- arthropodFeedsOnInsect
- animalEatsVegetable
- plantRepresentsEmotion
- foodDecreasesRiskOfDisease
- clothingGoesWithClothing
- bacteriaCausesPhysCondition
- buildingMadeOfMaterial
- emotionAssociatedWithDisease
- foodCanCauseDisease
- agriculturalProductAttractsInsect
- arteryArisesFromArtery
- countryHasSportsFans
- bakedGoodServedWithBeverage
- beverageContainsProtein
- animalCanDevelopDisease
- beverageMadeFromBeverage
Ontology Extension (2)

Goal:
• Add new subcategories

Approach:
• For each category C,
  • train NELL to read the relation
    SubsetOf\(_C\) : C \rightarrow C

*no new software here, just add this relation to ontology
NELL: subcategories discovered by reading

Animal:

• Pets
  – Hamsters, Ferrets, Birds, Dog, Cats, Rabbits, Snakes, Parrots, Kittens, …

• Predators
  – Bears, Foxes, Wolves, Coyotes, Snakes, Racoons, Eagles, Lions, Leopards, Hawks, Humans, …

Learned reading patterns for AnimalSubset(arg1,arg2)
"arg1 and other medium sized arg2"
"arg1 and other jungle arg2" "arg1 and other magnificent arg2" "arg1 and other pesky arg2" "arg1 and other mammals and arg2" "arg1 and other Ice Age arg2" "arg1 or other biting arg2" "arg1 and other marsh arg2" "arg1 and other migrant arg2" "arg1 and other monogastric arg2" "arg1 and other mythical arg2" "arg1 and other nesting
NELL: subcategories discovered by reading

Animal:
• Pets
  – Hamsters, Ferrets, Birds, Dog, Cats, Rabbits, Snakes, Parrots, Kittens, …
• Predators
  – Bears, Foxes, Wolves, Coyotes, Snakes, Raccoons, Eagles, Lions, Leopards, Hawks, Humans, …

Chemical:
• Fossil fuels
  – Carbon, Natural gas, Coal, Diesel, Monoxide, Gases, …
• Gases
  – Helium, Carbon dioxide, Methane, Oxygen, Propane, Ozone, Radon…

Learned reading patterns:
"arg1 and other medium sized arg2"  "arg1 and other jungle arg2"  "arg1 and other magnificent arg2"  "arg1 and other pesky arg2"  "arg1 and other mammals and arg2"  "arg1 and other Ice Age arg2"  "arg1 or other biting arg2"  "arg1 and other marsh arg2"  "arg1 and other migrant arg2"  "arg1 and other monogastric arg2"  "arg1 and other mythical arg2"  "arg1 and other nesting arg2"

Learned reading patterns:
"arg1 and other hydrocarbon arg2"  "arg1 and other aqueous arg2"  "arg1 and other hazardous air arg2"  "arg1 and oxygen are arg2"  "arg1 and such synthetic arg2"  "arg1 as a lifting arg2"  "arg1 as a tracer arg2"  "arg1 as the carrier arg2"  "arg1 as the inert arg2"  "arg1 as the primary cleaning arg2"  "arg1 and other noxious arg2"  "arg1 and other trace arg2"  "arg1 as the reagent arg2"  "arg1 as the tracer arg2"
NELL Architecture

Knowledge Base (latent variables)

- **Beliefs**
- **Candidate Beliefs**

Evidence Integrator

Input Sources

- **Text Context patterns (CPL)**
- **Orthographic classifier (CML)**
- **URL specific HTML patterns (SEAL)**
- **Human advice**

- **Actively search for web text (OpenEval)**
- **Infer new beliefs from old (PRA)**
- **Image classifier (NEIL)**
- **Ontology extender (OntExt)**
Key Idea 4: Cumulative, Staged Learning

Learning X improves ability to learn Y

1. Classify noun phrases (NP’s) by category
2. Classify NP pairs by relation
3. Discover rules to predict new relation instances
4. Learn which NP’s (co)refer to which latent concepts
5. Discover new relations to extend ontology
6. Learn to infer relation instances via targeted random walks
7. Vision: connect NELL and NEIL
8. Learn to microread single sentences
9. Learn to assign temporal scope to beliefs
10. Goal-driven reading: predict, then read to corroborate/correct
11. Make NELL a conversational agent on Twitter
12. Add a robot body to NELL
Consistency
Correctness
Self reflection
The core problem:
• Agents can measure internal *consistency*, but not *correctness*

Challenge:
• Under what conditions does *consistency* $\rightarrow$ *correctness*?
The core problem:
• Agents can measure internal \textit{consistency}, but not \textit{correctness}.

Challenge:
• Under what conditions does \textit{consistency} $\rightarrow$ \textit{correctness}?
• Can an autonomous agent determine its own accuracy?
Problem setting:

- have N different estimates $f_1, \ldots, f_N$ of target function $f^*$

\[
f_i : X \rightarrow Y; \quad Y \in \{0, 1\}
\]

- agreement between $f_i, f_j$:

\[
a_{ij} \equiv P_x(f_i(x) = f_j(x))
\]
Problem setting:
• have $N$ different estimates $f_1, \ldots, f_N$ of target function $f^*$
  \[ f_i : X \to Y; \; Y \in \{0, 1\} \]
• agreement between $f_i, f_j$: $a_{ij} \equiv P_x(f_i(x) = f_j(x))$

Key insight: errors and agreement rates are related

\[
a_{ij} = \Pr[\text{neither makes error}] + \Pr[\text{both make error}]
\]

\[
a_{ij} = 1 - e_i - e_j + 2e_{ij}
\]

- prob. $f_i$ and $f_j$ agree
- prob. $f_i$ error
- prob. $f_j$ error
- prob. $f_i$ and $f_j$ both make error

[Platanios, Blum, Mitchell, UAI 2014]
Estimating Error from Unlabeled Data

1. IF \( f_1, f_2, f_3 \) make indep. errors, and accuracies > 0.5
   THEN
   \[ a_{ij} = 1 - e_i - e_j + 2e_i e_j \]
   \[ a_{ij} = 1 - e_i - e_j + 2e_i e_j \]

Algorithm:
- use unlabeled data to estimate \( a_{12}, a_{13}, a_{23} \)
- solve three equations for three unknowns \( e_1, e_2, e_3 \)

measure errors from unlabeled data !!
Estimating Error from Unlabeled Data

1. IF $f_1, f_2, f_3$ make indep. errors, accuracies $> 0.5$
   THEN $a_{ij} = 1 - e_i - e_j + 2e_{ij}$
   $\Rightarrow a_{ij} = 1 - e_i - e_j + 2e_i e_j$

2. but if errors not independent
Estimating Error from Unlabeled Data

1. IF $f_1, f_2, f_3$ make indep. errors, accuracies $> 0.5$
   THEN $a_{ij} = 1 - e_i - e_j + 2e_{ij}$
   $\rightarrow a_{ij} = 1 - e_i - e_j + 2e_ie_j$

2. but if errors not independent

$\min (e_{ij} - e_ie_j)^2$

such that

$(\forall i, j) \ a_{ij} = 1 - e_i - e_j + 2e_{ij}$
True error (red), estimated error (blue)

NELL classifiers:

[Platanios, Blum, Mitchell, *UAI 2014*]
True error (red), estimated error (blue)

[Platanios, Blum, Mitchell, *UAI 2014*]

NELL classifiers:

Brain image fMRI classifiers:
Summary

1. Use *coupled* training for semi-supervised learning
2. Datamine the KB to learn probabilistic inference rules
3. Automatically extend ontology
4. Use staged learning curriculum

New directions:
- Self-reflection, self-estimates of accuracy
- Incorporate vision (NEIL)
- Microreading (Jayant Krishnamurthy)
- Assess pro/con evidence/arguments (Ndapa Nakashole)
- Use Twitter (Alan Ritter), news feeds
- Intelligent mobile agent (InMind project – Yahoo!)
thank you

and thanks to:
Darpa, Google, NSF, Yahoo!, Microsoft, Fulbright, Intel

follow NELL on Twitter:  @CMUNELL
browse/download NELL’s KB at http://rtw.ml.cmu.edu